# UNIVERSITAS GUNADARMA

# PRAKTIKUM KECERDASAN ARTIFICIAL

**MANUAL BOOK**

**"APPLYING NLP FOR ACCURATE LANGUAGE IDENTIFICATION IN TEXT DATA"**


| | |
|---|---|
| Nama | : Muhammad Rafi Ilham |
| Npm | : 51421034 |
| Kelas | : 3IA02 |
| Fakultas | : Teknologi Industri |
| Jurusan | : Informatika |
| PJ | : Hukama Shofhah |


Ditulis Guna Melengkapi Sebagian Syarat

Praktikum Kecerdasan Artificial

Universitas Gunadarma

**2023**

# DAFTAR ISI

# BAB I

# PENDAHULUAN

## 1.1 Latar Belakang

Pemrosesan Bahasa Alami atau *Natural Language Processing* (NLP) merupakan cabang dari kecerdasan bautan yang memiliki tujuan untuk memberi kemampuan pada komputer pemahaman dalam bentuk respon hasil Bahasa manusia secara alami. *Natural Language Processing* (NLP) menggabungkan beberapa jenis pembelajaran, seperti model statistik, machine learning, deep learning dan linguistik komputasi yang biasanya memproses data berupa teks dan audio dari manusia, dimana diharapkan dapat dipahami dengan makna dan sentimennya.

*Natural Language Processing* (NLP) mempunyai peran penting di pengembahan kecerdasan buatan, tujuannya memungkinkan komputer dapat memahami Bahasa manusia. *Natural Language Processing* (NLP) mempunyai solusi tantangan big data, yaitu dengan menyiapkan alat yang efektif untuk mengelola data tidak terstruktur. *Natural Language Processing* (NLP) punya manfaat untuk bisnis, seperti pendeteksian atau pemrosesan data besar dan analisis sentimen.

Pada kajian ini, saya akan menerapkan konsep NLP pada pendeteksian atau pengindetifikasian bahasa. *Natural Language Processing* (NLP) memiliki memanfaatkan teknik teknik untuk membuat komputer bisa memahami serta mengenali bahasa dalam sebuah *dataset* yang berupa teks. Melibatkan beberapa langkah, seperti tokenisasi, *stopword*, *lemmatization* atau *stemming*.

## 1.2 Tujuan

Penulis mengharapkan dengan ditulis nya manual ini dapat membantu untuk memberikan pemahaman mengenai pembuatan model dari cabang kecerdasan buatan (AI) yaitu *Natural Language Processing* (NLP) dan juga dalam penerapan nya pada sebuah pendeteksian Bahasa pada suatu *dataset* yang berupa suatu text, serta untuk memberikan pembaca panduan singkat dan praktis mengenai penerapan NLP untuk pendeteksian bahasa.

# BAB II

# PEMBAHASAN

## 2.1 Natural Language Processing (NLP)

*Natural Language Processing* (NLP) merupakan cabang dari kecerdasan buatan yang berfokus pada pemberian kemampuan kepada komputer untuk memahami, memproses, dan merespons bahasa manusia. Tujuannya untuk menciptakan sebuah sistem yang dapat berinteraksi dengan manusia melalui bahasa alami, baik berupa teks maupun ucapan.

*Natural Language Processing* (NLP) mencakup studi tentang bagaimana komputer dapat memahami makna dari teks dan meresponsnya dengan cara yang mirip dengan manusia. *Natural Language Processing* (NLP) memiliki Tujuan utama untuk menciptakan sistem yang dapat menerjemahkan bahasa manusia ke dalam representasi yang dapat dipahami oleh mesin, sehingga mesin dapat melakukan tugas tertentu berdasarkan instruksi atau permintaan bahasa manusia.

NLP bekerja dengan menganalisis dan memproses teks menggunakan teknik-teknik seperti tokenisasi, di mana teks dibagi menjadi unit-unit kecil seperti kata atau frasa, serta analisis sintaksis dan semantik untuk memahami struktur dan makna kalimat. Teknik seperti lemmatisasi dan stemming digunakan juga untuk mereduksi kata-kata ke bentuk dasar, sehingga memudahkan mesin memahami makna yang sebenarnya.

## 2.2 Teknik Pemrosesan Teks

*Natural Language Processing* (NLP) memiliki beberapa aspek utama yang memungkinkan untuk memudahkan atau mengefektifkan untuk pemahaman dan menganalisis teks manusia terdiri beberapa teknik umum, yaitu

- Tokenisasi

Tokenisasi ini melibatkan split teks atau pemecahan teks menjadi sebuah unit kecil yang disebut token, pada deteksi bahasa tokenisasi memungkinkan sistem untuk mengidentifikasi kata-kata atau frasa-frasa yang mungkin menjadi petunjuk kunci dalam menentukan bahasa yang digunakan.

- Lemmatisasi

  Merupakan proses pengubahan kata-kata ke bentuk dasarnya, dengan menggunakan teknik ini sistem dapat menormalisasi variasi kata yang mungkin muncul dalam berbagai bentuk, sehingga memudahkan pengenalan bahasa.

Penerapan teknik ini akan sangat berguna dalam deteksi bahasa, dimana mempersiapkan teks untuk analisis lebih lanjut, dengan peruraian teks sistem dapat lebih memprediksi menjadi lebih akurat dalam menentukan karakteristik dan pola bahasa.

### 2.3 Model dan Algoritma NLP

Model dan juga Algoritma NLP yang dapat digunakan dalam pendeteksian bahasa bisa bermacam macam, salah satu model yang biasa digunakan adalah algoritma Machine Learning seperti Logistic Regression, Naive bayes dan SVM. Model ini melibatkan tahap pelatihan pada data teks yang sudah dilabeli untuk mengidentifikasi pola dan hubungan antara teks dan bahasa yang ditargetkan.

Pemilihan model dan algoritma tergantung pada kompleksitas penugasan deteksi bahasa yang dihadapi dan ketersediaan data pelatihan yang sesuai. Pendekatan yang efektif dapat melibatkan kombinasi berbagai teknik ini untuk mencapai hasil yang optimal. Logistic Regression bekerja dengan baik dalam konteks ini dan dapat memberikan hasil yang memadai tergantung pada kualitas data dan kompleksitas tugas.

Pada pembuatan project untuk pendeteksian bahasa ini saya menggunakan pemodelan dengan Logistic Regression untuk melakukan sebuah klasifikasi bahasa. Model ini merupakan klasifikasi teks yang sederhana dan cukup efektif.

## 2.4 Aplikasi NLP dalam Deteksi Bahasa

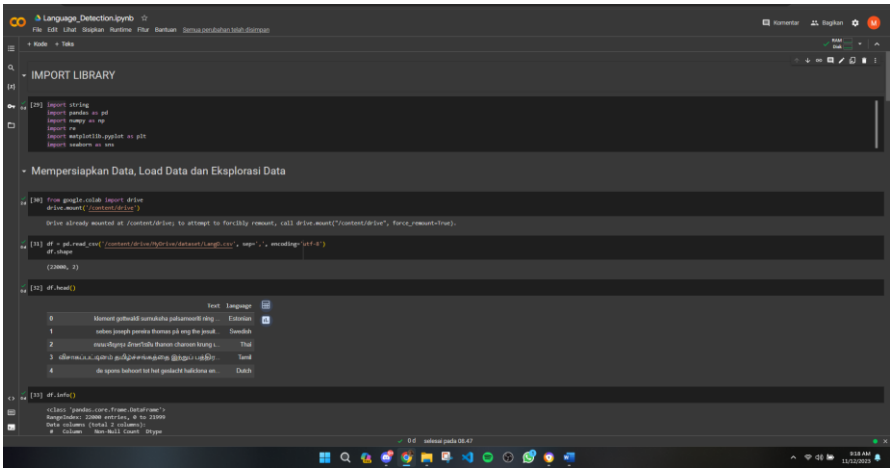
Pengaplikasian *Natural Language Processing* (NLP) dalam pendeteksian bahasa memainkan peran yang penting dalam berbagai konteks. Dengan menggunakan teknik seperti tokenisasi, penghapusan stopwords dan lemmatisasi bisa membuat NLP mengenali pola bahasa yang tidak biasa dalam teks dalam *dataset* nya.

Sistem mempunyai kemungkinan dalam mengidentifikasi dan mengklasifikasikan bahasa yang digunakan atau di uji cobakan. *Natural Language Processing* (NLP) digunakan untuk mendeteksi pola bahasa yang mungkin mengindikasikan niat atau sentimen tertentu yang berguna dalam analisis sentimen di media sosial atau yang lainnya. Pengaplikasikan NLP dalam deteksi bahasa membuka solusi inovatif dalam berbagai bidang, termasuk layanan pengguna.

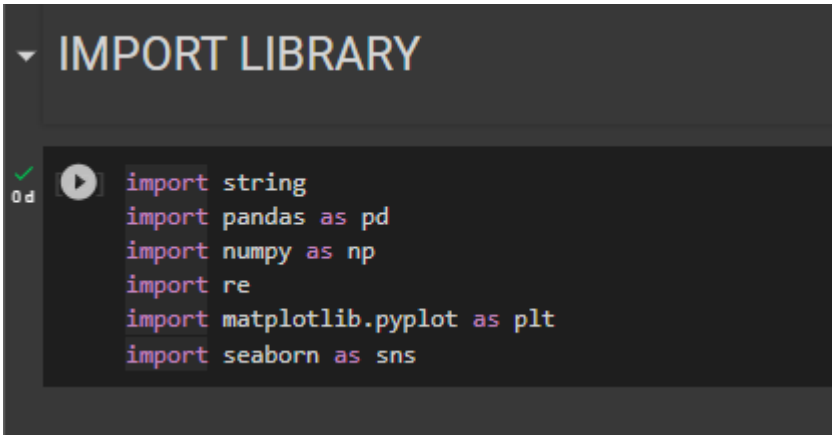
# BAB III

# ANALISA DAN PERANCANGAN

Pada Rancangan project untuk pengumpulan ujian mata praktikum kecerdasan artificial saya ini, saya menggunakan *tools* umum yang biasa sering digunakan dalam penganalisisan data dan lainnya yaitu google colab:

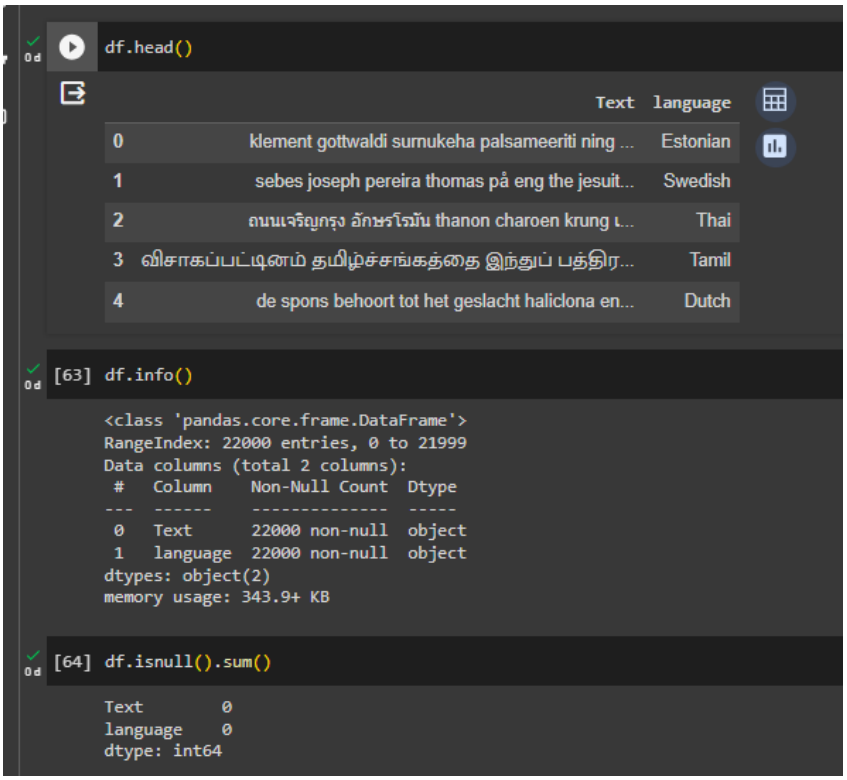



- Pada Langkah pertama, seperti pada umumnya kita perlu mempersiapkan hal hal yang kita butuhkan terlebih dahulu, sebelum melakukan percangan dan analisis lebih lanjut, yaitu dengan mengimport *library* umum yang dibutuhkan:

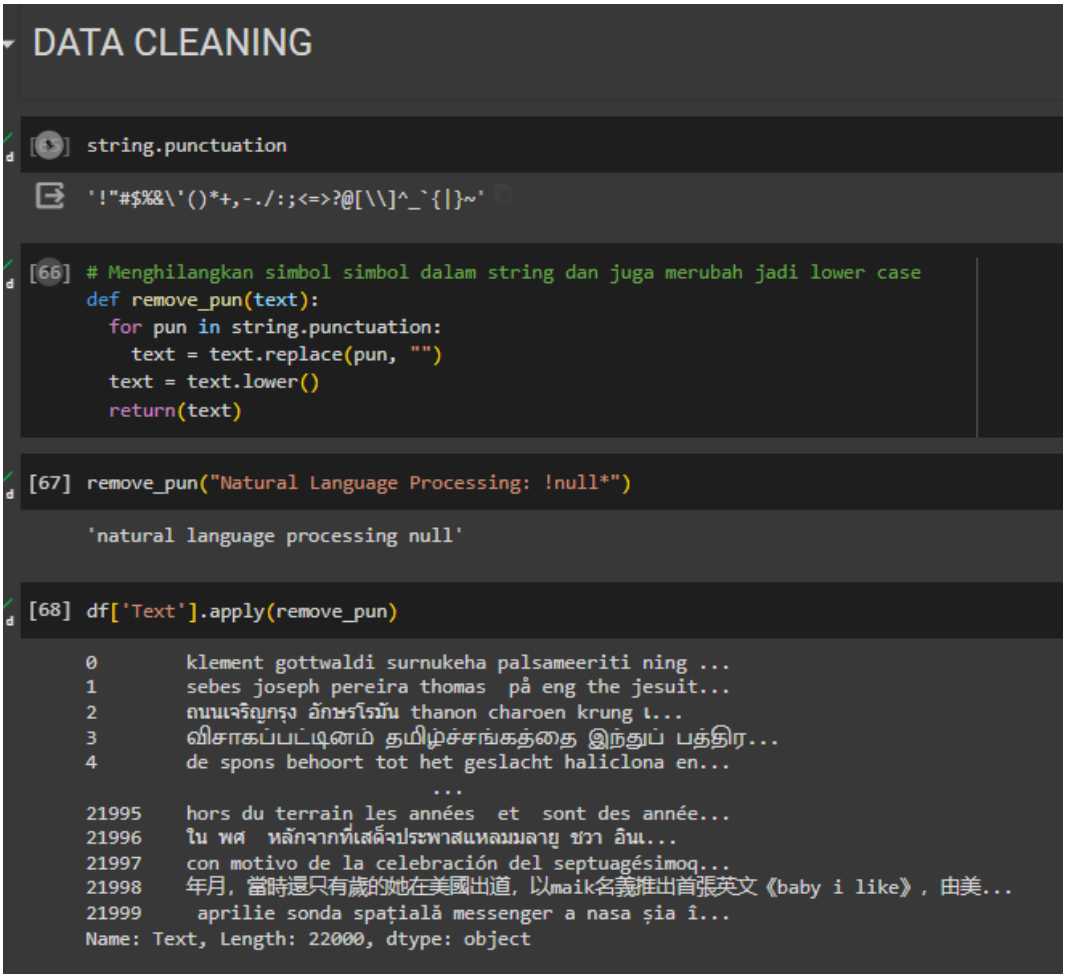


- Selanjutnya kita melakukan persiapkan data terlebih dahulu dengan melakukan pemuatan dataset, saya menggunakan konsep memuat data dengan melakukan *mount* pada google drive saya. Lakukan pembacaan file dataset pada *path* yang sesuai pada google drive. Lakukan pembacaan dimensi dari dataset untuk mengetahui dimensi dari dataset yang digunakan menggunakan .shape()

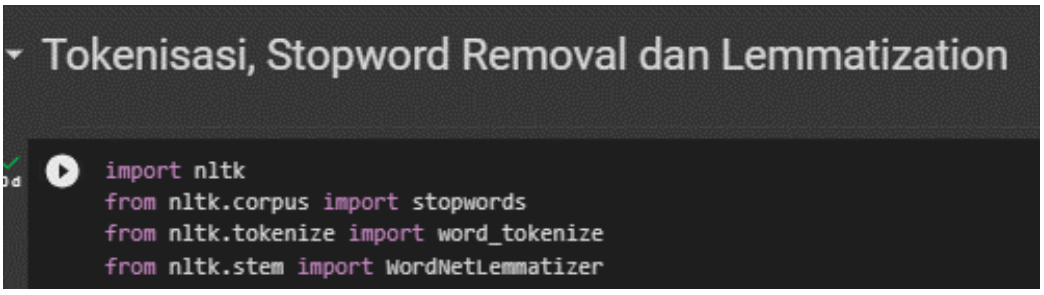

- Pada tahap selanjutnya masih melanjutkan tahapan dari sebelumnya yaitu akan mengeksplorasi data pada dataset, pada tahap ini saya menampilkan 5 baris pertama dari dataframe dan juga mengetahui informasi mengenai jumlah kolom dan kolom serta ada tidak nya data null atau hilang pada setiap kolom.



- Dikarenakan tidak adanya kolom null, maka tidak perlu pembersihan untuk menghilangkan kolom yang bernilai null pada umumnya, selanjutnya saya melakukan pembersihan data dengan membuat fungsi untuk menghilangkan simbol simbol dan membuat semua text menjadi *lowercase*. Lalu melakuan penampilan dari fungsi yang telah dibuat.




- Melakukan pengimportan untuk Tokenisasi, stopword removal dan lemmatization.



- Melakukan pengundudan *resource* untuk NLTK

```
# Download resource NLTK
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
```

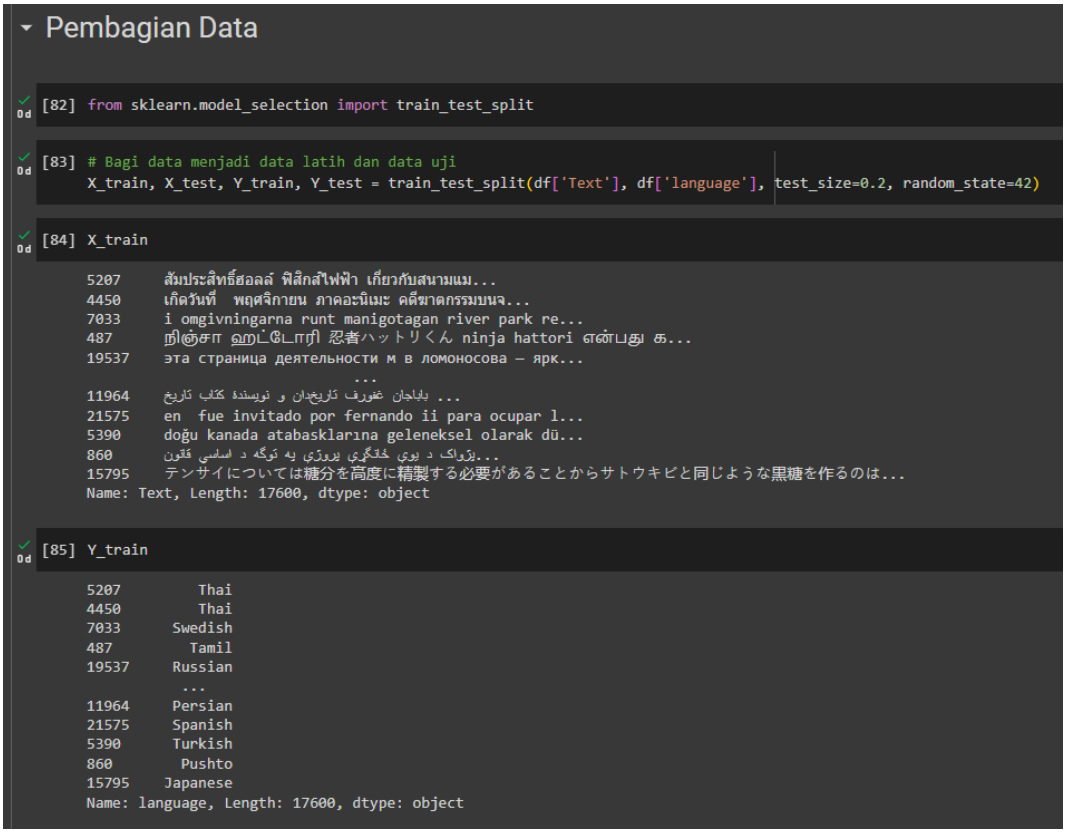
```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
True
```

- Melakukan tahap tokenisasi, stopword removal dan lemmatization dengan mengambil salah satu baris text pada dataset yaitu baris 43 atau index ke 42, melakukan tokenisasi pada text tersebut, lalu melakukan stopword removal pada hasil tokenisasi dan melakukan lematisasi dari hasil stopword removal. Lalu menampilkan hasil nya dengan cara melakukan print.

```
# Pilih salah satu baris dari dataset
sample_text = df['Text'].iloc[42]

# Tokenisasi
# Mecah -> token, untuk identifkasi frasa/kata
tokens = word_tokenize(sample_text)

# Stopword Removal
# Hilangkan kata generik/ tak khusus
stop_words = set(stopwords.words('english'))
filtered_tokens = [word for word in tokens if word.lower() not in stop_words]

# Lemmatization
# Generisasi ke bentuk dasar
lemmatizer = WordNetLemmatizer()
lemmatized_tokens = [lemmatizer.lemmatize(word) for word in filtered_tokens]

print("Original Text:", sample_text)
print("After Tokenization:", tokens)
print("After Stopword Removal:", filtered_tokens)
print("After Lemmatization:", lemmatized_tokens)
```

```
Original Text: lock helen  "a mans story is his gris-gris" ishmael reeds neo-hood
After Tokenization: ['lock', 'helen', '``', 'a', 'mans', 'story', 'is', 'his', 'g
After Stopword Removal: ['lock', 'helen', '``', 'mans', 'story', 'gris-gris', "''
After Lemmatization: ['lock', 'helen', '``', 'man', 'story', 'gris-gris', "''", '
```

- Selanjutnya melakukan pembagian data menjadi dua set, yaitu data untuk pelatihan (*training*) dan untuk pengujian (*testing*) menggunakan library scikit-learn. Serta menampilkan  data latih berlabel X dan Y.



Pembagian Data

```
[82] from sklearn.model_selection import train_test_split
```

```
[83] # Bagi data menjadi data latih dan data uji
     X_train, X_test, Y_train, Y_test = train_test_split(df['Text'], df['language'], test_size=0.2, random_state=42)
```

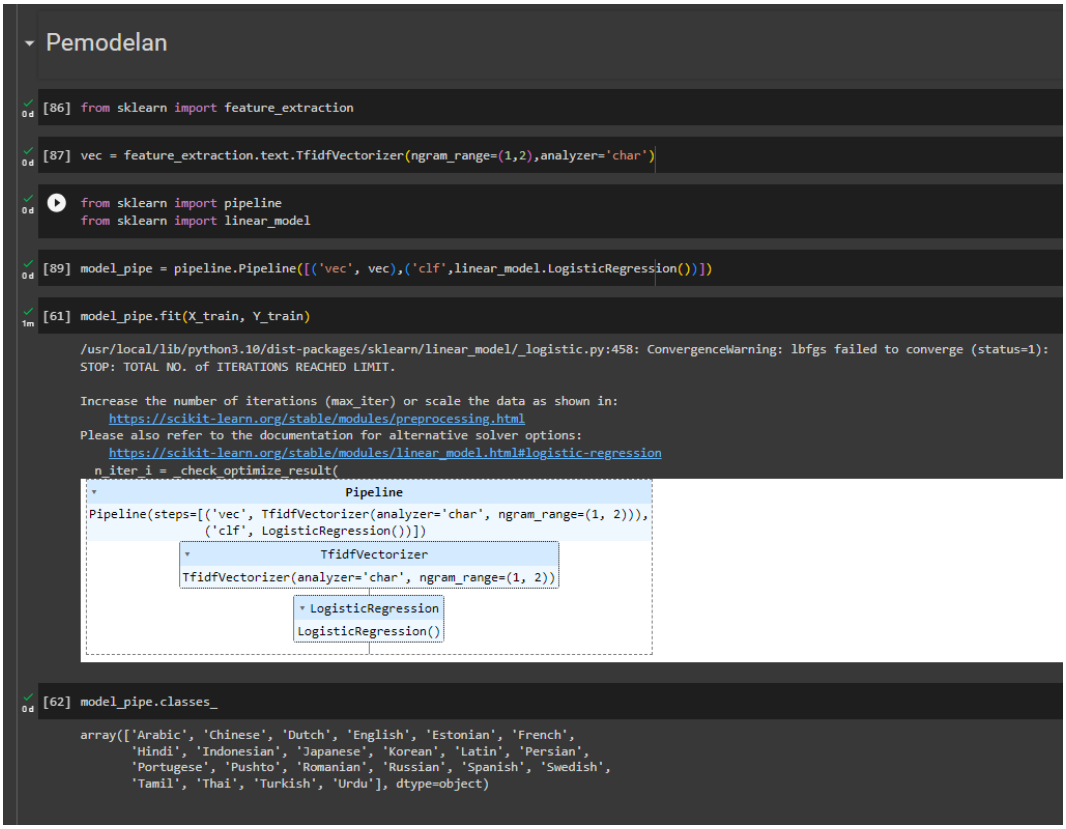
```
[84] X_train
```

```
5207     สัมประสิทธิ์ฮอลล์ ฟิสิกส์ไฟฟ้า เกี่ยวกับสนามแม...
4450     เกิดวันที่  พฤศจิกายน ภาคอะนิเมะ คดีฆาตกรรมบนจ...
7033     i omgivningarna runt manigotagan river park re...
487      நிஞ்சா ஹட்டோரி 忍者ハットリくん ninja hattori என்பது க...
19537    эта страница деятельности м в ломоносова — ярк...
                               ...
11964    باباجان غفورف تاریخدان و نویسندهٔ کتاب تاریخ ...
21575    en  fue invitado por fernando ii para ocupar l...
5390     doğu kanada atabasklarına geleneksel olarak dü...
860      بژواک د یوې ځانګړې پروژې په توګه د اساسي قانون...
15795    テンサイについては糖分を高度に精製する必要があることからサトウキビと同じような黒糖を作るのは...
Name: Text, Length: 17600, dtype: object
```

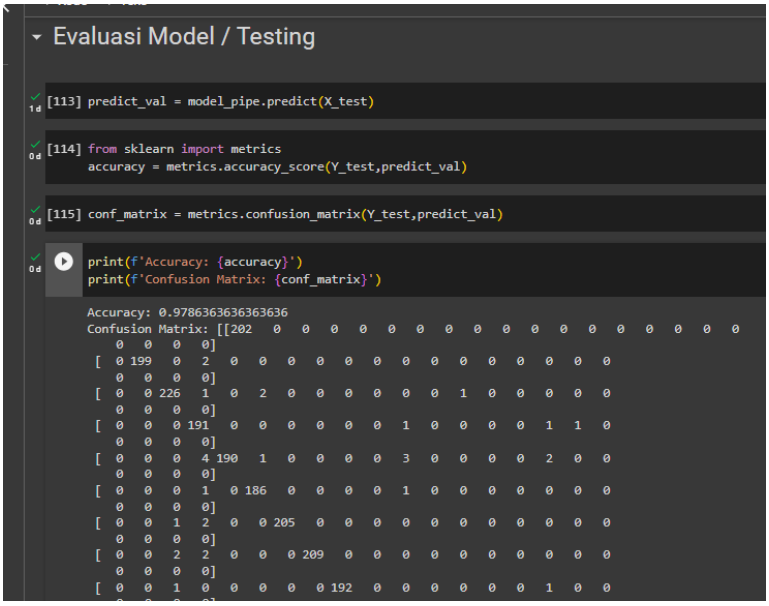
```
[85] Y_train
```

```
5207         Thai
4450         Thai
7033      Swedish
487         Tamil
19537     Russian
           ...
11964     Persian
21575     Spanish
5390      Turkish
860        Pushto
15795    Japanese
Name: language, Length: 17600, dtype: object
```

- Melakukan pemodelan dengan pendekatan TF-IDF Vectorization serta melakukan pengklasifikasikan menggunakan Logistic Regression serta melakukan pelatihan model dan terakhir menampilkan kelas kelas modelnya.



- Selanjutnya melakukan evaluasi model yang telah dibangun untuk mengukur seberapa baik model yang dibuat dapat mengkalsifikasikan bahasa pada data, dimana saya menggunakan cakupan akurasi dan matriks kebingungan.

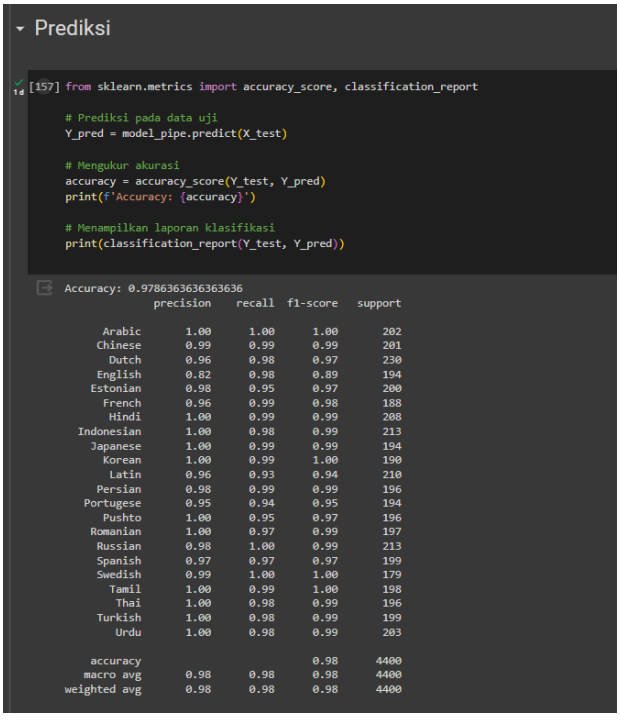


## Evaluasi Model / Testing

```
[113] predict_val = model_pipe.predict(X_test)

[114] from sklearn import metrics
      accuracy = metrics.accuracy_score(Y_test,predict_val)

[115] conf_matrix = metrics.confusion_matrix(Y_test,predict_val)

      print(f'Accuracy: {accuracy}')
      print(f'Confusion Matrix: {conf_matrix}')
```

```
Accuracy: 0.9786363636363636
Confusion Matrix: [[202   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0]
 [  0 199   0   2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0]
 [  0   0 226   1   0   2   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0
    0   0   0   0]
 [  0   0   0 191   0   0   0   0   0   0   1   0   0   0   0   1   1   0   0   0   0
    0   0   0   0]
 [  0   0   0   4 190   1   0   0   0   0   3   0   0   0   0   2   0   0   0   0   0
    0   0   0   0]
 [  0   0   0   1   0 186   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0
    0   0   0   0]
 [  0   0   1   2   0   0 205   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0]
 [  0   0   2   2   0   0   0 209   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0]
 [  0   0   1   0   0   0   0   0 192   0   0   0   0   0   0   1   0   0   0   0   0
```

- Melakukan prediksi lagi untuk model yang telah dilatih pada data uji dan menambahkan untuk mengukur akurasi model pada data uji yang diklasifikasikan untuk memberi laporan dan pemahaman.

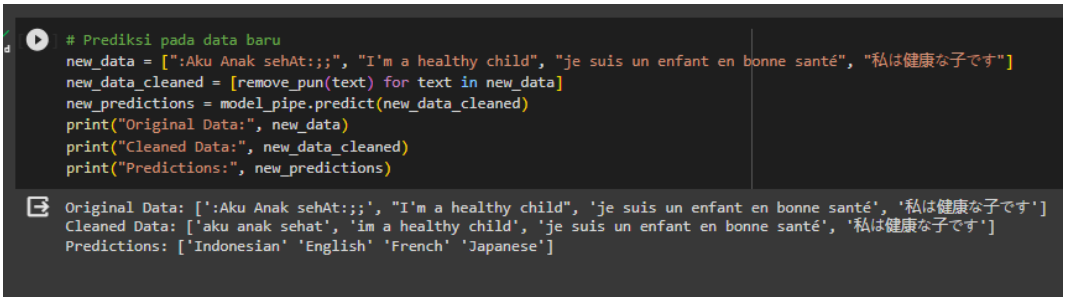


## Prediksi

```
[157] from sklearn.metrics import accuracy_score, classification_report

      # Prediksi pada data uji
      Y_pred = model_pipe.predict(X_test)

      # Mengukur akurasi
      accuracy = accuracy_score(Y_test, Y_pred)
      print(f'Accuracy: {accuracy}')

      # Menampilkan laporan klasifikasi
      print(classification_report(Y_test, Y_pred))
```

```
Accuracy: 0.9786363636363636
              precision    recall  f1-score   support

      Arabic       1.00      1.00      1.00       202
     Chinese       0.99      0.99      0.99       201
       Dutch       0.96      0.98      0.97       230
     English       0.82      0.98      0.89       194
    Estonian       0.98      0.95      0.97       200
      French       0.96      0.99      0.98       188
       Hindi       1.00      0.99      0.99       208
  Indonesian       1.00      0.98      0.99       213
    Japanese       1.00      0.99      0.99       194
      Korean       1.00      0.99      1.00       190
       Latin       0.96      0.93      0.94       210
     Persian       0.98      0.99      0.99       196
   Portugese       0.95      0.94      0.95       194
      Pushto       1.00      0.95      0.97       196
    Romanian       1.00      0.97      0.99       197
     Russian       0.98      1.00      0.99       213
     Spanish       0.97      0.97      0.97       199
     Swedish       0.99      1.00      1.00       179
       Tamil       1.00      0.99      1.00       198
        Thai       1.00      0.98      0.99       196
     Turkish       1.00      0.98      0.99       199
        Urdu       1.00      0.98      0.99       203

    accuracy                           0.98      4400
   macro avg       0.98      0.98      0.98      4400
weighted avg       0.98      0.98      0.98      4400
```

- Melakukan tahap prediksi pada data baru dengan menampilkan data original nya, yang sudah dibersihkan serta prediksi dengan memperlihatkan label languages nya.



```
# Prediksi pada data baru
new_data = [":Aku Anak sehAt:;;", "I'm a healthy child", "je suis un enfant en bonne santé", "私は健康な子です"]
new_data_cleaned = [remove_pun(text) for text in new_data]
new_predictions = model_pipe.predict(new_data_cleaned)
print("Original Data:", new_data)
print("Cleaned Data:", new_data_cleaned)
print("Predictions:", new_predictions)
```

```
Original Data: [':Aku Anak sehAt:;;', "I'm a healthy child", 'je suis un enfant en bonne santé', '私は健康な子です']
Cleaned Data: ['aku anak sehat', 'im a healthy child', 'je suis un enfant en bonne santé', '私は健康な子です']
Predictions: ['Indonesian' 'English' 'French' 'Japanese']
```

# BAB IV

# PENUTUP

## 4.1 Kesimpulan

Penerapan Natural Language Processing (NLP) untuk pendeteksian bahasa dalam proyek ini menunjukkan potensi besar dalam memberikan kemampuan komputer untuk memahami dan mengenali bahasa manusia. Langkah-langkah analisis teks, seperti tokenisasi, penghapusan stopword, dan lemmatisasi, membuktikan pentingnya persiapan data sebelum dilibatkan dalam pembuatan model. Algoritma Logistic Regression dipilih sebagai pendekatan klasifikasi teks yang sederhana namun efektif. Proses pembelajaran mesin dan evaluasi model menyoroti pentingnya kualitas data, parameter optimal, dan pemilihan model yang sesuai.

## 4.2 Saran

Untuk pengembangan lebih lanjut, perlu diperhatikan peningkatan kualitas data dengan label yang akurat dan representasi yang seimbang dari setiap kelas bahasa. Eksplorasi model NLP lainnya seperti Naive Bayes atau SVM dapat memberikan wawasan tambahan tentang kinerja algoritma. Disarankan untuk melakukan optimasi parameter pada model Logistic Regression dan TF-IDF Vectorizer serta mempertimbangkan teknik embedding kata untuk meningkatkan pemahaman bahasa.

Penanganan bahasa lain dan penanganan ketidakseimbangan kelas juga merupakan poin penting yang perlu diperhatikan. Jika dataset mencakup berbagai bahasa, proyek dapat diperluas untuk menangani lebih dari satu bahasa dengan menggunakan teknik transfer learning atau mengumpulkan lebih banyak data untuk setiap bahasa. Selain itu, penanganan ketidakseimbangan kelas dengan teknik oversampling atau undersampling dapat meningkatkan kinerja model pada bahasa yang kurang diwakili. Dengan mempertimbangkan saran-saran ini, proyek ini memiliki potensi untuk meningkatkan kinerja dan relevansinya dalam deteksi bahasa.